



Europäisches Patentamt  
European Patent Office  
Office européen des brevets



(11) Publication number:

**0 417 013 A2**

(12)

## EUROPEAN PATENT APPLICATION

(21) Application number: 90402475.9

(51) Int. Cl.<sup>5</sup>: G06F 9/30

(22) Date of filing: 07.09.90

(30) Priority: 08.09.89 JP 231705/89

(36) Date of publication of application:  
13.03.91 Bulletin 91/11

(46) Designated Contracting States:  
DE FR GB

(71) Applicant: FUJITSU LIMITED  
1015, Kamikodanaka Nakahara-ku  
Kawasaki-shi Kanagawa 211(JP)

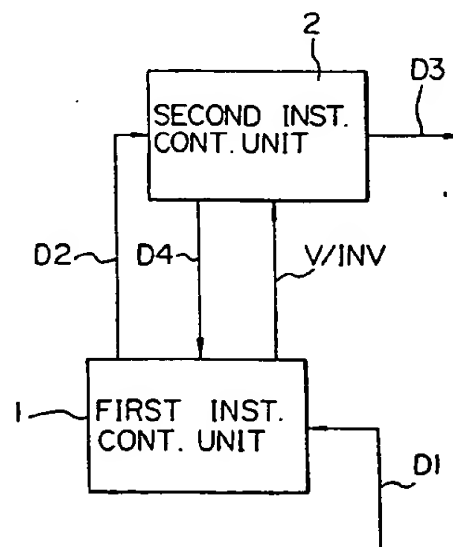
(72) Inventor: Yoshitake, Akihiro  
1-7-1, Kamishinjo, Nakahara-ku  
Kawasaki-shi, Kanagawa 211(JP)  
Inventor: Ohshima, Toshiharu  
c/o Mr. Aoki, 997-2, Chitose, Takatsu-ku  
Kawasaki-shi, Kanagawa 213(JP)

(74) Representative: Joly, Jean-Jacques et al  
CABINET BEAU DE LOMENIE 55, rue  
d'Amsterdam  
F-75008 Paris(FR)

(54) Data processor decoding and executing a train of instructions of variable length.

(57) A data processor decoding and executing a train of instructions of variable length. The data processor includes a first instruction control means (1) for temporarily storing a prefetched instruction code (D1) and sequentially outputting said instruction code with units of a predetermined number of bits, and a second instruction control means (2) for decoding an instruction code (D2) fed from the first instruction control means, generating control information (D3) for data processing based on the decoding, and outputting data (D4) indicating instruction update demand quantity to the first instruction control means. On the basis of the data indicating the requested quantity of update instructions, the first instruction control means judges whether it has output a valid instruction code of length exceeding the update demand quantity, and provides an indication (V/INV) of validity or invalidity of the decoded instruction code and controls updating of the instruction code based on a result of the judgement. As a result, it becomes possible to reduce time necessary for the supply of instruction codes and thus improve a data processing speed as the entire processor.

*Fig. 1*



EP 0 417 013 A2

## DATA PROCESSOR DECODING AND EXECUTING A TRAIN OF INSTRUCTIONS OF VARIABLE LENGTH

### BACKGROUND OF THE INVENTION

#### 1. Field of the Invention

The present invention relates to a data processor, more particularly, to a microprocessor which decodes and executes a train of instructions of variable length.

#### 2. Description of the Related Art

A typical microprocessor includes an instruction buffer for storing prefetched instruction codes, and a decoder which receives a train of instructions of variable length fed from the instruction buffer and decodes them to produce control information for data processing (e.g., control information on pipeline control, microprogram addresses, or the like). To decode and execute the train of instructions of variable length, the decoder must receive valid data, i.e., valid instruction codes. To this end, the instruction buffer in the conventional processor predecodes the train of instructions to always supply the decoder with valid data. At this time, the instruction buffer determined an update quantity of the instruction to be next decoded by the decoder.

However, where the instruction buffer must predecode many bits among instruction codes to determine update quantity of instructions, the required predecoding time is as long time as that necessary for the decoding of instructions by the decoder. Accordingly, a drawback occurs in that the instruction buffer cannot quickly feed valid instruction codes to the decoder. This leads to a lowering in data processing speed as the entire processor and thus is not preferable.

### SUMMARY OF THE INVENTION

An object of the present invention is to provide a data processor which can increase its data processing speed as a whole.

According to the present invention, there is provided a data processor decoding and executing a train of instructions of variable length, the data processor including: a first instruction control unit for temporarily storing a prefetched instruction code and sequentially outputting the instruction code with units or a predetermined number of bits; and a second instruction control unit, operatively connected to the first instruction control unit, for decoding an instruction code fed from the first

instruction control unit, generating control information for data processing based on the decoding, and outputting data indicating requested quantity of update instructions to the first instruction control unit, wherein the first instruction control unit effects a judgement, based on the data indicating the requested quantity of update instructions, of whether it has output a valid instruction code of length exceeding the demanded update quantity, and provides an indication of validity or invalidity of the decoded instruction code and controls updating of the instruction code based on a result of the judgement.

### BRIEF DESCRIPTION OF THE DRAWINGS

Other objects and features of the present invention will be described hereinafter in detail by way of preferred embodiments with reference to the accompanying drawings, in which:

Fig. 1 is a block diagram illustrating the principle of the data processor according to the present invention;

Fig. 2 is a functional block diagram illustrating an entire constitution of a microprocessor according to an embodiment of the present invention;

Fig. 3 is a block diagram illustrating a constitution of the instruction buffer shown in Fig. 2;

Fig. 4 is a block diagram illustrating a constitution of the decoder shown in Fig. 2;

Fig. 5 is a circuit diagram illustrating a constitution of the main part of the instruction buffer of Fig. 3;

Fig. 6 is a circuit diagram illustrating a constitution of the main part of the decoder of Fig. 4;

Fig. 7 is a timing chart illustrating an example of the operation of the embodiment of Fig. 2; and Figs. 8a to 8e are views illustrating instruction formats employed in the embodiment of Fig. 2.

### DESCRIPTION OF THE PREFERRED EMBODIMENTS

Figure 1 illustrates the principle of the data processor according to the present invention.

Referring to Fig. 1, the data processor according to the present invention, which decodes and executes a train of instructions of variable length, includes a first instruction control unit 1 for temporarily storing a prefetched instruction code D1 and sequentially outputting the instruction code

with units of a predetermined number of bits, and a second instruction control unit 2 for decoding an instruction code D2 fed from the first instruction control unit, generating control information D3 for data processing based on the decoding, and outputting data D4, hereafter termed "update demand quantity", indicative of the requested quantity of update instructions to the first instruction control unit.

In instances where the first instruction control unit judges, data D4 indicating the requested quantity of update instructions, that it has output a valid instruction code of length exceeding the update demand quantity, it provides an indication - (in) that the decoded instruction code is valid, and updates instruction codes. On the other hand, in instances, where the first instruction control unit judges that it has not output a valid instruction code of length exceeding the update demand quantity, it provides an indication (INV) that the decoded instruction code is invalid, and causes the second instruction control unit to effect the decoding again on the identical stage.

In the above constitution, the first instruction control unit 1 outputs an instruction code irrespective of that the instruction code is valid or invalid. The second instruction control unit 2 in turn assumes that the instruction code D2 fed from the first instruction control unit 1 is valid, and outputs the update demand quantity D4. Based on the update demand quantity D4, the first instruction control unit 1 judges validity or invalidity of the decoding and controls updating of the instruction code.

Therefore, it becomes unnecessary to effect predecoding for determination of update quantity as seen in the prior art and thus it becomes possible to reduce time necessary for the supply of instruction codes. This contributes to an improvement in data processing speed as the entire processor.

Next, a preferred embodiment of the present invention will be explained with reference to Figs. 2 to 8e.

Figure 2 illustrates an entire constitution of the microprocessor as an embodiment of the present invention.

The microprocessor according to the present embodiment includes an instruction control unit IC having an instruction buffer 11, a decoder 12, a pipeline controller 13, and a microprogram 14; an instruction execution unit IE having an address generator 15, an operation unit 16, and a register file 17; a memory control unit MC having an instruction access controller 18 and an operand access controller 19; and a bus control unit BC having an address controller 20, a bus supervisory controller 21, and a data transmitting and receiving

unit 22. The instruction control unit IC reads, decodes, and executes instructions. The constitutions and operations of the instruction execution unit IE, memory control unit MC and bus control unit BC are not directly related to the present invention and thus the explanation thereof is omitted.

Next, the constitution of the instruction buffer 11 and decoder 12 in the instruction control unit IC will be explained with reference to Figs. 3 and 4, respectively. Note, the instruction buffer 11 and decoder 12 correspond to the first instruction control unit 1 and second instruction control unit 2 shown in Fig. 1, respectively.

Referring to Fig. 3, the instruction buffer 11 includes a buffer memory 31 of 8 bytes (B) x 6 in capacity; an input buffer 33 interposed between an instruction cache 32 of the instruction access controller 18 and the buffer memory 31; a rotator 34 for rotating bits to bring a first bit of an instruction code read out of the buffer memory 31 to a first bit of an iQ bus QB; an instruction write/read controller 35 for controlling write/read operation of instructions (codes) for the buffer memory 31; a validity judging unit 36 for judging the validity of a decoded instruction; an instruction prefetch controller 37 for controlling fetching of an instruction code from the instruction cache 32; and an error detector 38 for selecting an exception processing based on an output of the instruction prefetch controller 37.

The instruction buffer 11 prefetches instruction codes from the instruction cache 32 into the buffer memory 31 until the buffer memory 31 becomes full of instruction codes. The fetched codes are sequentially read out with units of 16 bits via the iQ bus QB to the decoder 12. When the decoding of each 16-bit unit is completed, the buffer memory 31 becomes empty and in turn a new prefetch operation is carried out. In the read operation from the buffer memory 31, the rotator 34 rotates an instruction code read out of the buffer memory 31 with units of 16 bits, thereby bringing a first bit of an instruction code to be next decoded to a first bit position of the iQ bus QB.

The instruction code of 16-bit unit fed to the decoder 12 from the instruction buffer 11 via the iQ bus QB constitutes part of a train of instructions of variable length having formats, for example, as shown Figs. 8a to 8e. Note, in Figs. 8a to 8e, reference OPD denotes an operand designating part; reference EX an extended part; reference OP an instruction code part; reference R a register designating part; reference S an operand size designating part; reference #I an immediate value; and reference disp a displacement part.

Next, referring to Fig. 4, the decoder 12 includes a first instruction decoding unit 41; a second instruction decoding unit 42; an addressing decoding unit 43; an update demand quantity decoding

unit 44 for computing an update demand quantity of instruction codes; an added mode decoding unit 45; a decoding sequencer 46 for forming a current decoding stage of instruction codes and sequentially selecting outputs of each of the decoding units according to the decoding stage; a selector 47 for selecting one of outputs of the first and second instruction decoding units 41 and 42 under the control of the decoding sequencer 46; a register input unit 48 for receiving data from the iQ bus QB; a register output unit 49 for passing an output of the register input unit 48 in response to an internal clock  $\phi 3$  fed from the decoding sequencer 46; latches 41a, 42a, 43a, 43b, 44a, and 45a disposed in the input sides of the decoding units and responsive to an internal clock  $\phi 1$  fed from the decoding sequencer 46; and latches 41b, 42b, 44b, and 45b disposed in the output sides of the decoding units and responsive to the internal clock  $\phi 3$ .

The decoder 12 receives a train of instructions of variable length (see Figs. 8a to 8e) fed through the iQ bus QB, and decodes them on the basis of 16-bit base length. Concretely, the decoder 12 decodes the type and addressing mode, or the like, of a supplied instruction, and generates control information such as pipeline control information, operand-address computing information and micro-program addresses that are used for data processing. At the same time, the decoder 12 outputs data RQHW indicating update demand quantity of instruction codes to the instruction buffer 11.

Based on the update demand quantity data RQHW, the instruction buffer 11 effects a validity-invalidity judgement (DCOVALiD) of the decoding and controls updating of instruction codes.

Figure 5 illustrates a constitution of the main part of the instruction buffer 11, i.e., the validity judging unit 36.

In the illustration, references iQVO3 to iQVO0 denote signals indicating how many half-words (HWs) counted from an upper bit are valid among 64-bit data (63 to 0) output to the iQ bus. For example, where upper two half-words (bits 63 to 32) are valid, the signals iQVO0 and iQVO1 are at level "H", and the signals iQVO2 and iQVO3 are at level "L". References RQHW4 to RQHW1 denote update demand quantity indication signals (instruction code demand signals) mentioned above. For example, when the decoder 12 demands two half-words to be updated, only the signal RQHW2 is at level "H", and the other signals are at level "L". Reference iQWaiT denotes a wait demand signal; references HW0 to HW4 denote signals indicating update quantity of instruction codes that must be updated by the decoding; and reference DCOVALiD denotes a signal indicating the validity or invalidity of the decoding.

The circuit of Fig. 5 includes AND gates 500 to

503 responding to the wait demand signal iQWaiT and the signals iQVO0 to iQVO3; latch circuits 510 to 513 responding to a signal  $\phi 3$  (an inverted signal  $\phi 3X$ ) to latch outputs of the AND gates 500 to 503; NAND gates 520 to 523 responding to output signals of the latch circuits 510 to 513 and instruction code demand signals RQHW1 to RQHW4; inverters 530 to 533 of input inverting type responding to outputs of the NAND gates 520 to 523 and providing the signals HW1 to HW4; inverters 540 to 543 responding to outputs of the latch circuits 510 to 513; an AND gate 51 responding to an output of the inverter 543 and the instruction code demand signal RQHW4; an AND gate 52 responding to an output of the inverter 542 and the instruction code demand signal RQHW3; a NOR gate 53 responding to outputs of the AND gates 51 and 52; an AND gate 54 responding to an output of the inverter 541 and the instruction code demand signal RQHW2; a NOR gate 55 responding to an output of the AND gate 54 and an output of the inverter 540; an OR gate 56 of input inverting type responding to outputs of the NOR gates 53 and 55 to provide the signal HW0; and an inverter 57 responding to an output of the OR gate 56 to provide the signal DCOVALiD for judging the validity of decoded instruction codes.

As explained above, the instruction buffer 11 produces the information iQVO3 to iQVO0 indicating how many HWs counted from the upper bit are valid among data output to the iQ bus QB, based on valid bits of the respective buffer. Based on the produced information and the instruction code demand signals RQHW4 to RQHW1 fed from the decoder 12, the validity judging unit 36 in the instruction buffer 11 judges whether or not instruction codes must be updated. Concretely, where the instruction buffer 11 has output a valid instruction code of length exceeding the update demand quantity, the instruction buffer 11 provides a validity judging signal DCOVALiD of "H" level indicating that the decoded instruction code is valid, and effects updating of instruction codes. On the other hand, where the instruction buffer 11 has not output a valid instruction code of length exceeding the update demand quantity, it provides a validity judging signal DCOVALiD of "L" level indicating that the decoded instruction code is invalid. In this case, the decoder 12 effects the decoding again on the identical stage.

Figure 6 illustrates a constitution of the main part of the decoder 12 of Fig. 4. The illustration corresponds to the update demand quantity decoding unit 44, part of the decoding sequencer 46, and latches provided in the periphery thereof.

The circuit of Fig. 6 includes a programmable logic array (PLA) 600; a NAND gate 601 responding to a signal  $\phi 1$  and a decoding input latch

enable signal DCiLE1; an inverter 602 of input inverting type responding to an output of the NAND gate 601; a latch circuit 603 for latching an instruction code from the iQ bus QB in response to the control signal  $\phi 1$  (an inverted signal  $\phi 1X$ ) and supplying the same to the PLA 600; a NAND gate 604 responding to the signal  $\phi 1$  and a decoding input latch enable signal DCiLE2; an inverter 605 of input inverting type responding to an output of the NAND gate 604; a latch circuit 606 for latching an instruction code from the iQ bus QB in response to the control signal  $\phi 1$  (inverted signal  $\phi 1X$ ) and supplying the same to the PLA 600; a NAND gate 611 responding to a signal  $\phi 3$  and a decoding output latch enable signal DCLE; an inverter 612 of input inverting type responding to an output of the NAND gate 611; latch circuits 613 to 616 for latching data from the PLA 600 in response to the signal 3 (inverted signal  $\phi 3X$ ) and providing the instruction code demand signals RQHW4 to RQHW1; latch circuits 617 to 619 for latching data from the PLA 600 in response to the signal  $\phi 3$  (inverted signal  $\phi 3X$ ); a NAND gate 621 responding to the signal  $\phi 1$  and the signal DCOVALiD indicating the validity of decoded instruction codes; an inverter 622 of input inverting type responding to an output of the NAND gate 621; latch circuits 623 to 625 for latching outputs of the latch circuits 617 to 619 in response to the signal  $\phi 1$  (inverted signal  $\phi 1X$ ) and providing decoding signals DC1 to DC3; latch circuits 626 and 627 for latching outputs of the latch circuits 623 and 624 in response to the control signal  $\phi 3$  (inverted signal  $\phi 3X$ ); an inverter 630 responding to the validity judging signal DCOVALiD; an AND gate 631 responding to an output of the latch circuit 617 and the validity judging signal DCOVALiD; an AND gate 632 responding to an output of the latch circuit 626 and an output of the inverter 630; a NOR gate 633 responding to outputs of the AND gates 631 and 632; an inverter 634 of input inverting type for generating a decoding latch enable signal DCiLE1 in response to an output of the NOR gate 633; an AND gate 635 responding to an output of the latch circuit 618 and the validity judging signal DCOVALiD; an AND gate 636 responding to an output of the latch circuit 627 and an output of the inverter 630; a NOR gate 637 responding to outputs of the AND gates 635 and 636; and an inverter 638 of input inverting type for generating a decoding latch enable signal DCiLE2 in response to an output of the NOR gate 637.

Next, the operation of the microprocessor according to the present embodiment will be explained with reference to the timing chart of Fig. 7, citing an example of a MOV (move) instruction.

#### (a) Cycle ①

The decoder 12 latches data (instruction codes) from the iQ bus QB in response to the control signal  $\phi 1$ , and decodes the same through the PLA 600. In the example of Fig. 7, a result of the decoding tells that three half-words each comprising 16 bits can simultaneously be decoded. The decoder 12 asserts, therefore, the instruction code demand signal RQHW3 to request updating of instruction codes of three half-words. On the other hand, the instruction buffer 11 produces the information iQVO3 to iQVO0 indicating how many HWs counted from the upper bit are valid among data output to the iQ bus QB, based on valid bits of the respective buffer. According to the produced information and the instruction code demand signals RQHW4 to RQHW1 fed from the decoder 12, the validity judging unit 36 (see Fig. 5) in the instruction buffer 11 judges whether or not instruction codes must be updated.

In this case, since the instruction buffer 11 has not output a valid instruction code of length exceeding the update demand quantity, it negates the validity judging signal DCOVALiD, asserts the instruction code HW0, and outputs the instruction code from the identical location of the buffer to the iQ bus QB in the next cycle ②.

Since the validity judging signal DCOVALiD has been negated, the decoder 12 recognizes (detects) that the decoded instruction code is invalid, by means of the circuit of Fig. 6, and effects the decoding again on the identical stage.

#### (b) Cycle ②

A decoding processing similar to that of Cycle ① is carried out. The instruction buffer 11 outputs instruction codes prefetched in the cycle ① to the iQ bus QB (bits 31 to 0). As a result, valid codes of four half-words are ready, and the validity judging signal DCOVALiD is asserted. The instruction buffer 11 updates instruction codes of three half-words and the decoder 12 in turn transits its operation to the next decoding stage DC2.

#### (c) Cycle ③

The decoder 12 provides an update demand quantity of two half-words. There is, however, valid data of only one half-word, so that the validity judging signal DCOVALiD is negated.

#### (d) Cycle ④

There are two valid half-words. However, a wait demand signal iQWaiT is provided under the pipeline control, so that the signals iQVO1 to iQVO3 are masked to negate the validity judging signal DCOVALiD.

(e) Cycle ⑤

Since the wait demand signal iQWaiT is negated, the validity judging signal DCOVALiD is asserted. The instruction buffer 11 updates instruction codes of two half-words and the decoder 12 in turn transits its operation to the next decoding stage DC1 (Cycle ⑥).

As explained above, according to the present embodiment, the instruction buffer 11 sequentially feeds instruction codes with units of a predetermined number of bits to the decoder 12, irrespective of the validity or invalidity of the instruction codes. The decoder 12 in turn assumes that the instruction codes fed from the instruction buffer 11 are valid, and outputs update demand quantity to the instruction buffer 11. Based on the update demand quantity, the instruction buffer 11 judges validity or invalidity of the decoding and controls updating of the instruction code. Accordingly, it becomes unnecessary to effect predecoding for determination of update quantity as seen in the prior art and thus it becomes possible to reduce time necessary for the supply of instruction codes. This results in an improvement in data processing speed as the entire processor.

Although the present invention has been disclosed and described by way of one embodiment, it is apparent to those skilled in the art that other embodiments and modifications of the present invention are possible without departing from the spirit or essential features thereof.

### Claims

1. A data processor decoding and executing a train of instructions of variable length, said data processor comprising:

a first instruction control means (1) for temporarily storing a prefetched instruction code (D1) and sequentially outputting said instruction code with units of a predetermined number of bits; and

a second instruction control means (2), operatively connected to said first instruction control means, for decoding an instruction code (D2) fed from the first instruction control means, generating control information (D3) for data processing based on the decoding, and outputting data (D4) indicating a requested quantity of update instructions to the first instruction control means,

wherein said first instruction control means (1) effects a judgement, based on the data (D4) indicating the requested quantity of update instructions, on whether it has output a valid instruction code of length exceeding the requested quantity of update instructions, and provides an indication (V/INV) of validity or invalidity of the decoded instruction code and controls updating of the instruction code based on a result of the judgement.

2. A data processor as set forth in claim 1, wherein, when the first instruction control means (1) judges that it has output a valid instruction code of length exceeding the requested quantity of update instructions, it provides an indication (V) that the decoded instruction code is valid, and updates the instruction code, and when the first instruction control means judges that it has not output a valid instruction code of length exceeding the requested quantity of update instructions, it provides an indication (INV) that the decoded instruction code is invalid, and causes the second instruction control means to effect the decoding again on the identical stage.

3. A data processor as set forth in claim 2, wherein, said first instruction control means comprises means (31) for storing the prefetched instruction code, means (34) for controlling arrangement of bits of the instruction code to sequentially feed the instruction code with units of the predetermined number of bits to said second instruction control means, and means (36) for judging the validity or invalidity of the instruction code decoded by the second instruction control means, whereby the first instruction control means supplies the second instruction control means with data (DCOVALiD) indicating validity or invalidity of the decoding.

4. A data processor as set forth in claim 2, wherein said second instruction control means comprises means (44) for computing the requested quantity of update instructions, means (41,42,43,45) for decoding instruction code information, addressing information and additional information included in the train of instructions of variable length, and a control means (46) for forming a current decoding stage of instruction codes and sequentially selecting each of the outputs of the computing means and decoding means according to the decoding stage, whereby the second instruction control means supplies the first instruction control means with data (RQHW) indicating the requested quantity of update instructions.

5. A data processor as set forth in claim 4, wherein said second instruction control means further comprises means (41a-5a, 41b-45b), responsive to control clocks ( $\phi 1$ ,  $\phi 3$ ) from said control means, for latching each information to be input to the computing means and decoding means and each information output from said computing means and decoding means.

6. A data processor as set forth in claim 5, wherein said computing means and decoding means are constituted by a programmable logic array (600).

5

10

15

20

25

30

35

40

45

50

55

7

*Fig. 1*

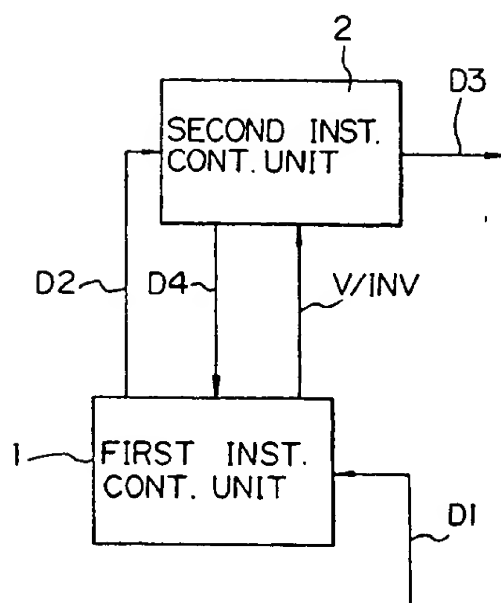




Fig. 2

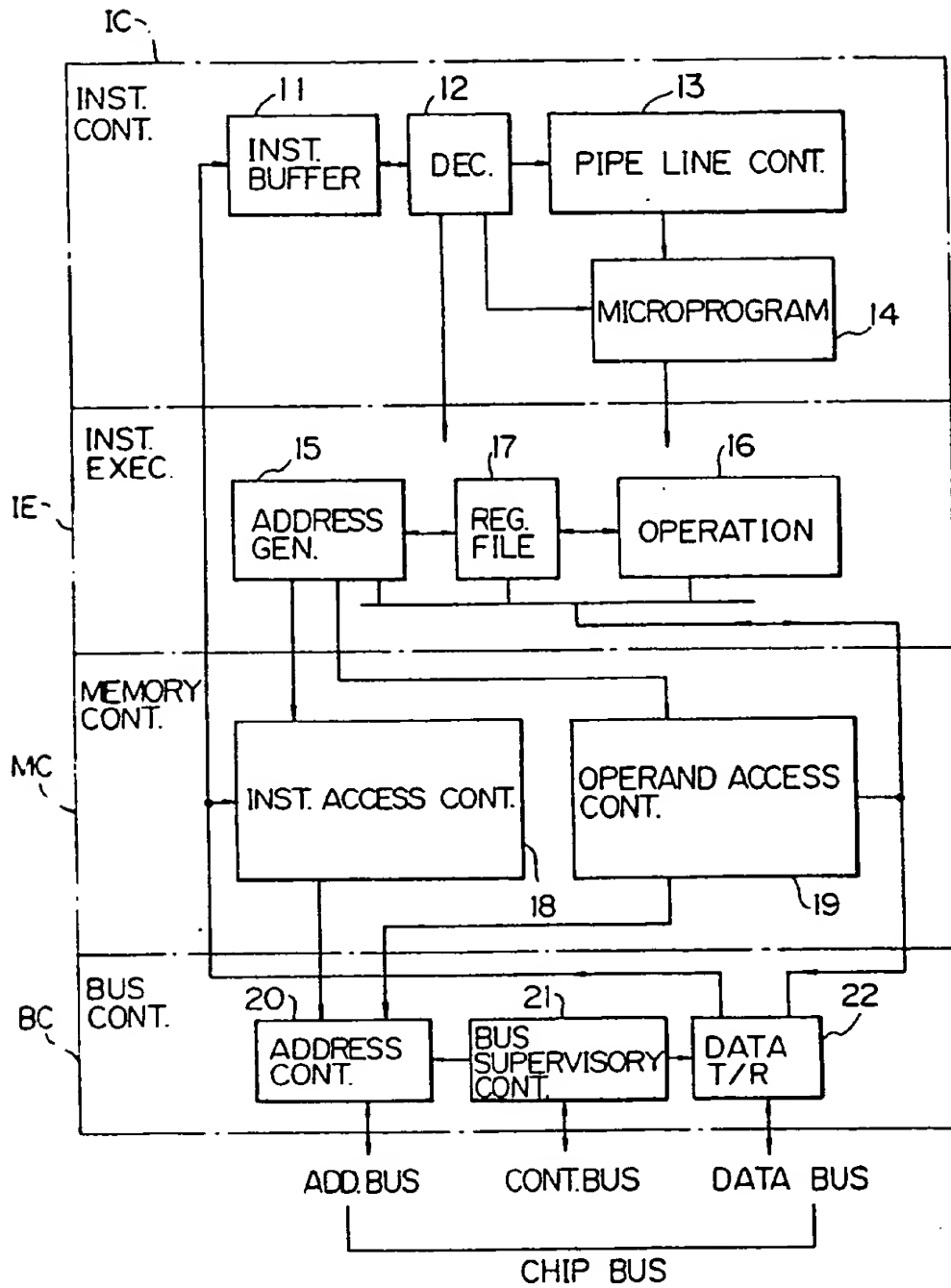


Fig.3

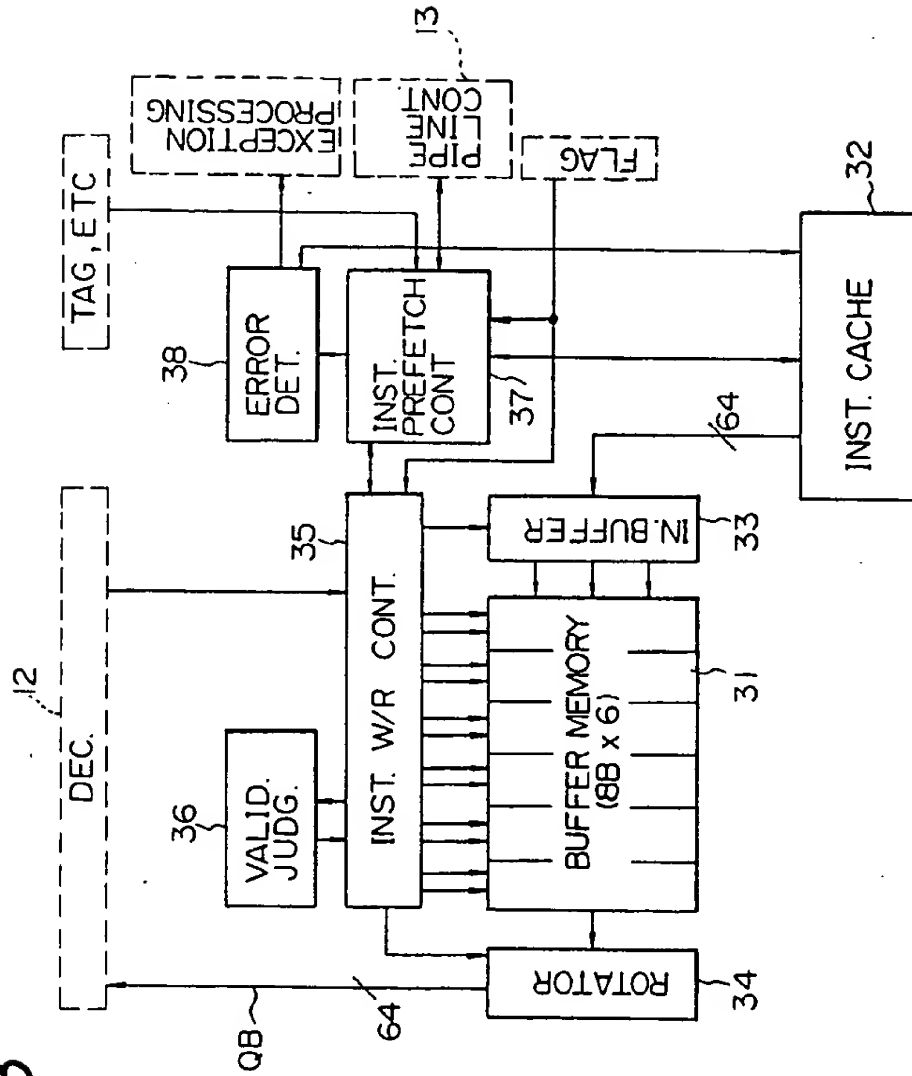


Fig.4

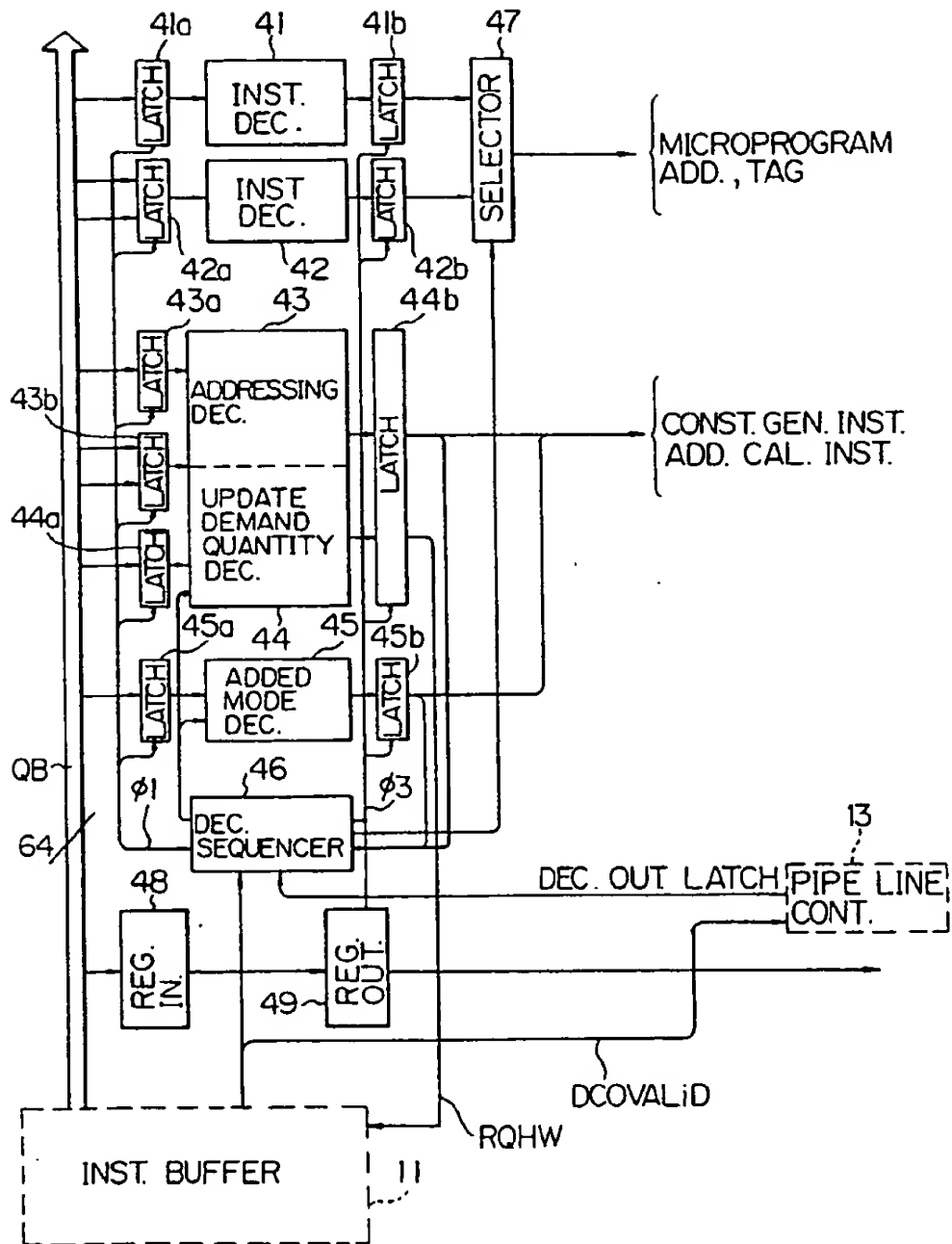


Fig. 5

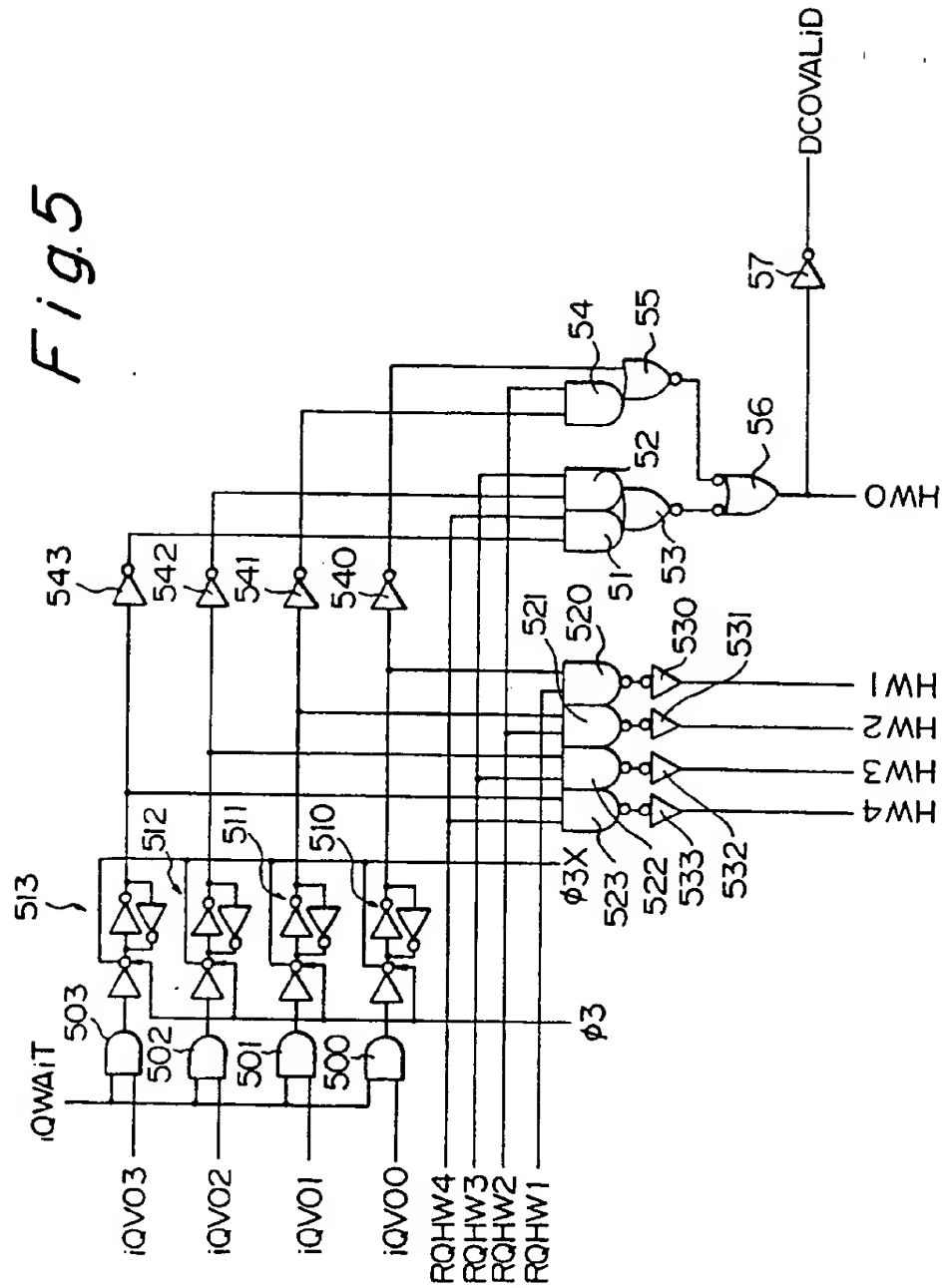


Fig. 6

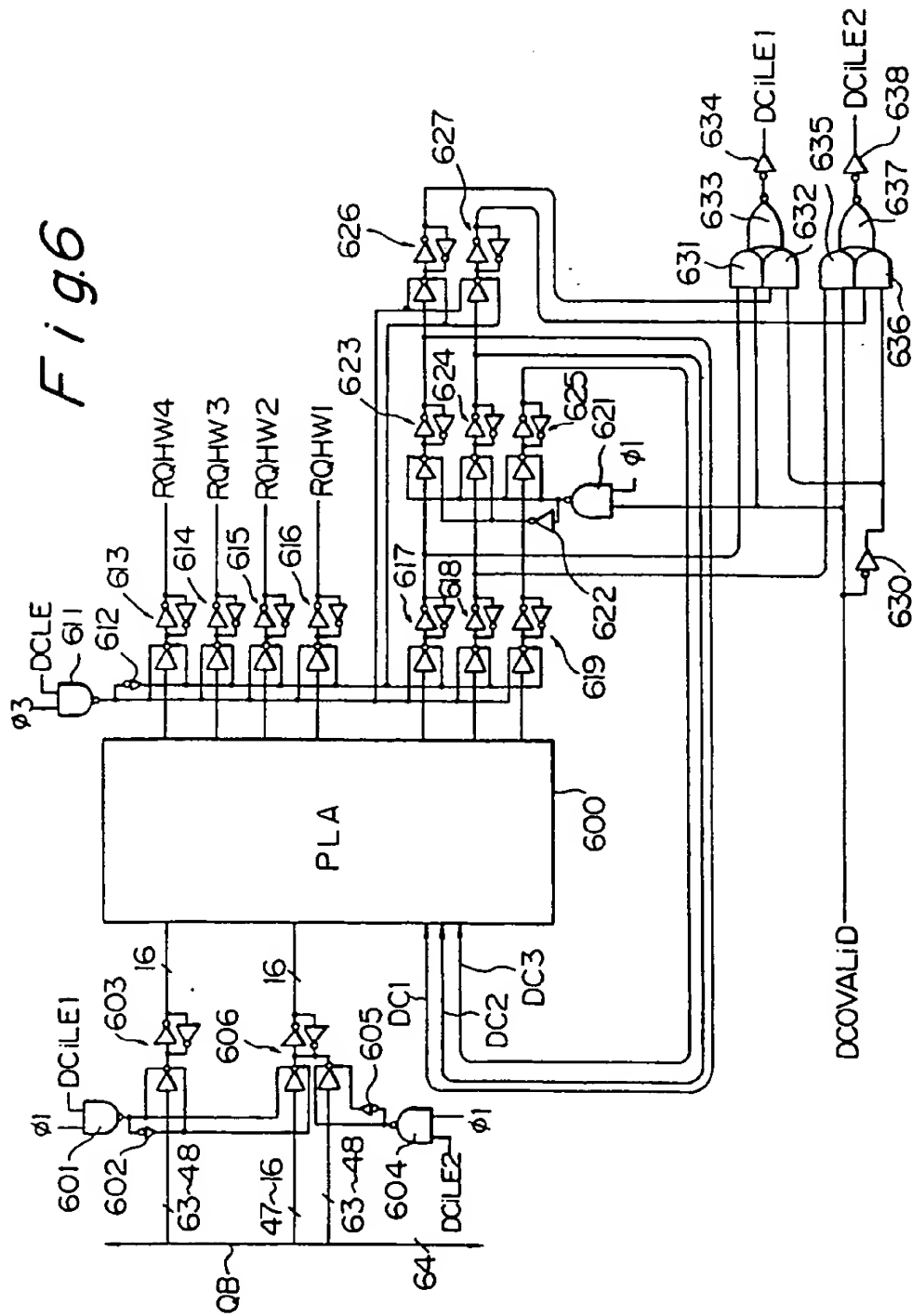
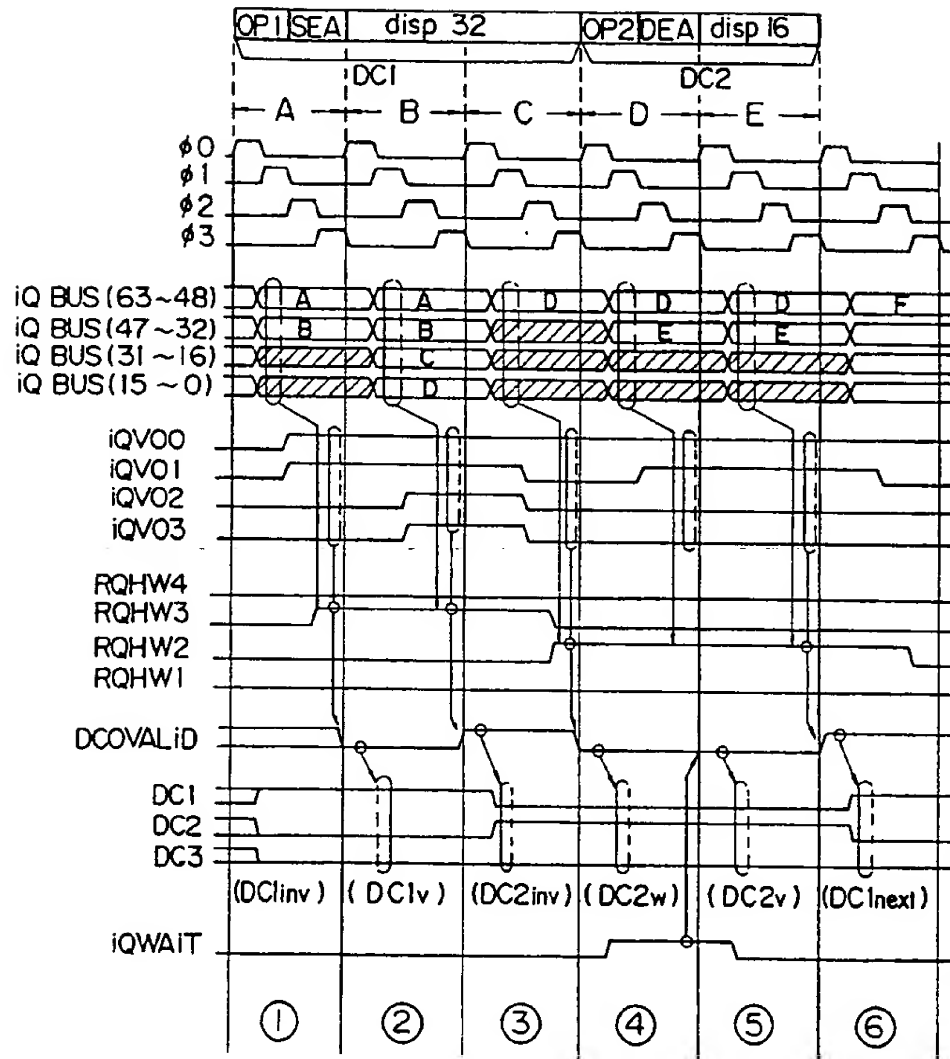
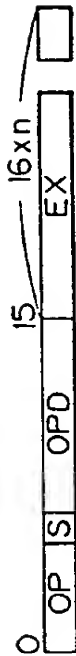


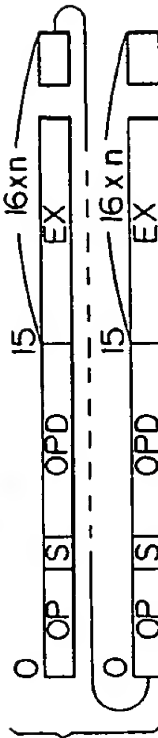
Fig.7



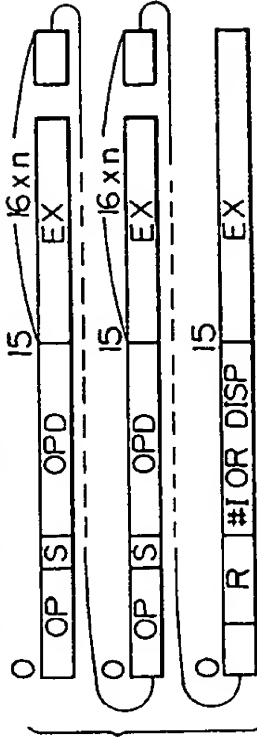
*Fig. 8a*  
(ONE OPERAND)



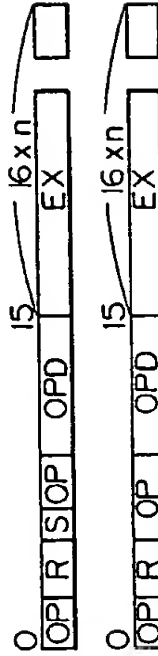
*Fig. 8b*  
(TWO OPERANDS)



*Fig. 8c*  
(EXTENDED OPERAND)



*Fig. 8d*  
(MEMORY-REG.)



*Fig. 8e*  
(REG.-REG.)

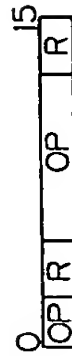
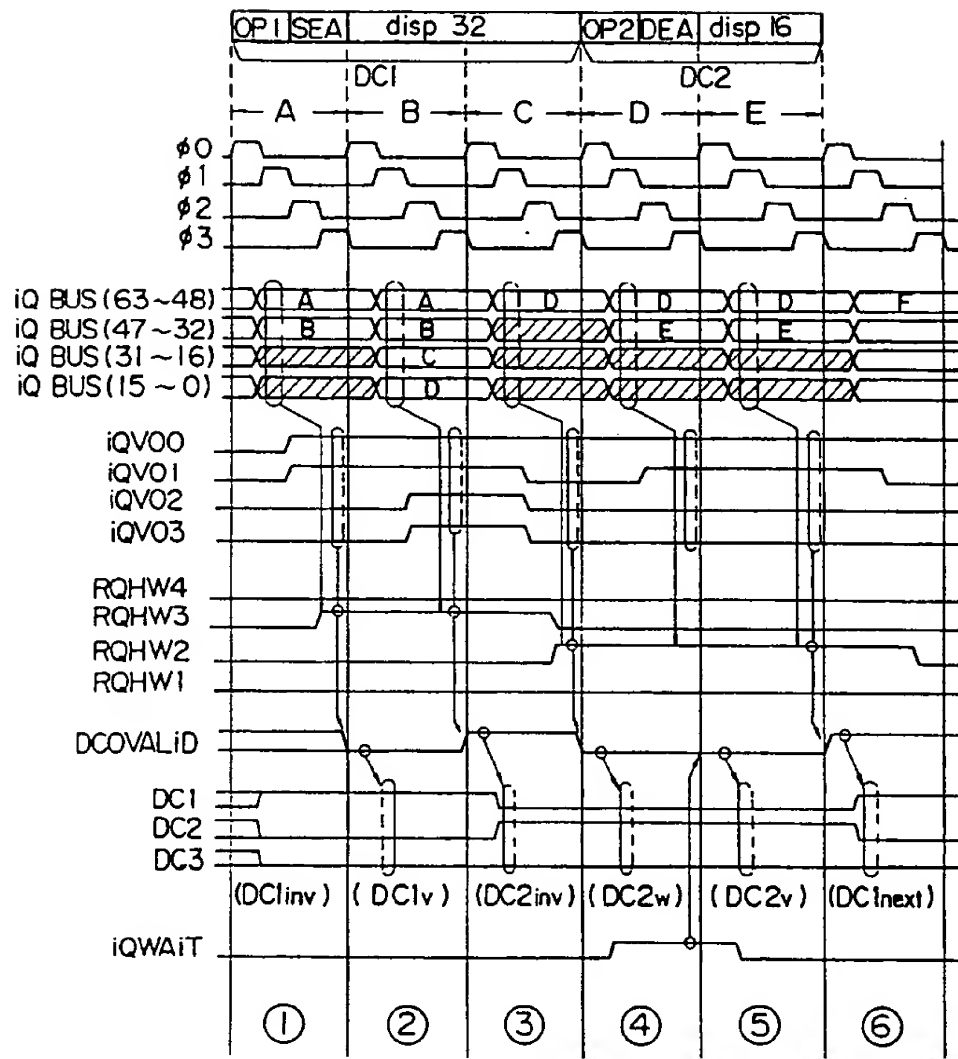
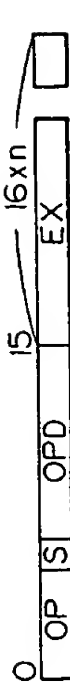


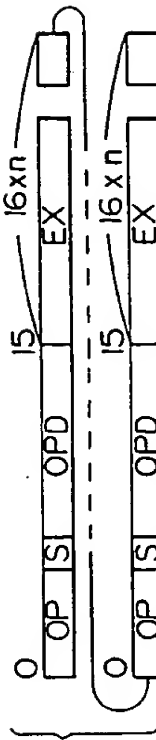
Fig. 7



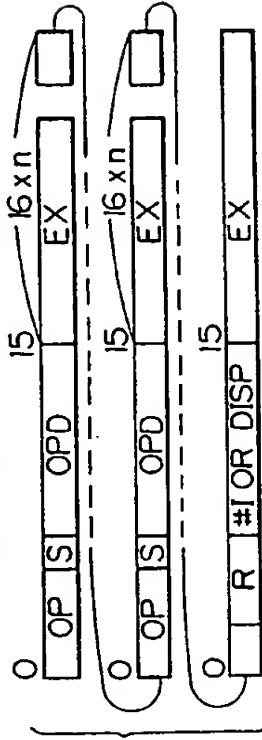




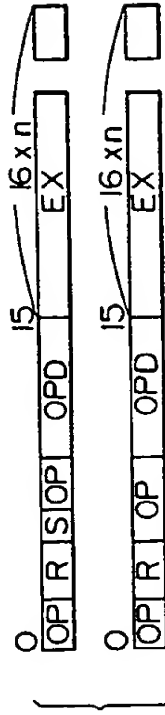
*Fig. 8a*  
(ONE OPERAND)



*Fig. 8b*  
(TWO OPERANDS)



*Fig. 8c*  
(EXTENDED OPERAND)



*Fig. 8d*  
(MEMORY-REG.)



*Fig. 8e*  
(REG.-REG.)

**THIS PAGE BLANK (USPTO)**



Europäisches Patentamt  
European Patent Office  
Office européen des brevets



Publication number: **0 417 013 A3**

(12)

## EUROPEAN PATENT APPLICATION

(21) Application number: 90402475.9

(51) Int. Cl.<sup>5</sup>: G06F 9/30, G06F 9/38

(22) Date of filing: 07.09.90

(30) Priority: 08.09.89 JP 231705/89

(43) Date of publication of application:  
13.03.91 Bulletin 91/11

(84) Designated Contracting States:  
DE FR GB

(88) Date of deferred publication of the search report:  
21.04.93 Bulletin 93/16

(71) Applicant: FUJITSU LIMITED  
1015, Kamikodanaka Nakahara-ku  
Kawasaki-shi Kanagawa 211(JP)

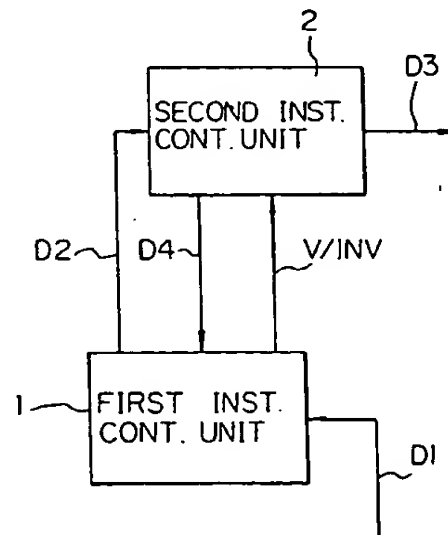
(72) Inventor: Yoshitake, Akihiro  
1-7-1, Kamishinjo, Nakahara-ku  
Kawasaki-shi, Kanagawa 211(JP)  
Inventor: Ohshima, Toshiharu  
c/o Mr. Aoki, 997-2, Chitose, Takatsu-ku  
Kawasaki-shi, Kanagawa 213(JP)

(74) Representative: Joly, Jean-Jacques et al  
Cabinet Beau de Loménie 158, rue de  
l'Université  
F-75340 Paris Cédex 07 (FR)

(54) Data processor decoding and executing a train of instructions of variable length.

(57) A data processor decoding and executing a train of instructions of variable length. The data processor includes a first instruction control means (1) for temporarily storing a prefetched instruction code (D1) and sequentially outputting said instruction code with units of a predetermined number of bits, and a second instruction control means (2) for decoding an instruction code (D2) fed from the first instruction control means, generating control information (D3) for data processing based on the decoding, and outputting data (D4) indicating instruction update demand quantity to the first instruction control means. On the basis of the data indicating the requested quantity of update instructions, the first instruction control means judges whether it has output a valid instruction code of length exceeding the update demand quantity, and provides an indication (V/INV) of validity or invalidity of the decoded instruction code and controls updating of the instruction code based on a result of the judgement. As a result, it becomes possible to reduce time necessary for the supply of instruction codes and thus improve a data processing speed as the entire processor.

*Fig. 1*



EP 0 417 013 A3



European Patent  
Office

## EUROPEAN SEARCH REPORT

Application Number

EP 90 40 2475

DOCUMENTS CONSIDERED TO BE RELEVANT			
Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int. Cl.5)
P,Y	EP-A-0 380 854 (DIGITAL EQUIPMENT CORP.) 8 August 1990 * column 20, line 41 - column 22, line 3; figures 5-8 *	1	G06F9/30 G06F9/38
P,Y	EP-A-0 352 103 (DIGITAL EQUIPMENT CORP.) 24 January 1990 * page 7, line 22 - line 26 * * page 9, line 31 - line 36 * * page 10, line 22 - line 26 *	1	
A	IEEE TRANSACTIONS ON COMPUTERS vol. C-33, no. 8, August 1984, NEW YORK US pages 713 - 730 B. W. LAMPSON ET AL. 'An Instruction Fetch Unit for a High-Performance Personal Computer' * page 726, right column *	1	
A	US-A-4 236 206 (STRECKER ET AL.) 25 November 1980		
			TECHNICAL FIELDS SEARCHED (Int. Cl.5)
			G06F
The present search report has been drawn up for all claims			
Place of search THE HAGUE		Date of completion of the search 22 FEBRUARY 1993	Examiner DASKALAKIS T.
CATEGORY OF CITED DOCUMENTS			
X : particularly relevant if taken alone Y : particularly relevant if combined with another document of the same category A : technological background O : non-written disclosure P : intermediate document		T : theory or principle underlying the invention E : earlier patent document, but published on, or after the filing date D : document cited in the application L : document cited for other reasons * : member of the same patent family, corresponding document	

EPO FORM 1503 (1.12.92) (P0401)